



Nota de implementación

- Debe enviar exactamente un archivo (el nombre del archivo esta dado en el enunciado de la tarea).
- Este archivo implementa el subprograma descrito en el enunciado de la tarea usando las firmas (declaraciones de funciones/métodos) dados en los archivos ejemplo.
- Estos subprogramas deben comportarse como se describe en el enunciado de la tarea.
- Es libre de implementar otros subprogramas (funciones, procedimientos, métodos).
- Sus envíos no deben interactuar de ninguna manera con la entrada/salida estándar, ni con archivos. En particular, si su programa imprime algo a cualquier flujo de salida, obtendrá el resultado SV (Security Violation). Puede escribir cualquier cosa al flujo de salida de error estándar.

Convenciones

Los enunciados de las tareas y los detalles de implementación en las secciones usan algunos tipos de nombres genericos en particular:

- El nombre de *array* y el tipo correspondiente de dato `int[]`
- El tipo de dato `int64`
- El tipo de dato `string`
- El tipo de dato `boolean`

En cada uno de los lenguajes de programación soportados, los "Graders" usan los tipos de datos apropiados del lenguaje como corresponde a la lista de abajo:

Language	array	int64	string	boolean
C++	<code>std::vector<int></code>	<code>long</code> <code>long</code>	<code>std::string</code>	<code>bool</code>
C	<code>int*</code>	<code>long</code> <code>long</code>	<code>char*</code>	<code>int</code>
Pascal	<code>array of longint</code>	<code>int64</code>	<code>string</code>	<code>boolean</code>
Java	<code>int[]</code>	<code>long</code>	<code>String</code>	<code>boolean</code>

Limites

Problem	Time Limit	Memory Limit
---------	------------	--------------

Problem	Time Limit	Memory Limit
Paint By Numbers	2 seconds	2 GB
Unscrambling a Messy Bug	2 seconds	2 GB
Aliens	2 seconds	2 GB